文本领域的数据预处理技术以及深度学习训练技巧

汇报人：扣扣

大纲

- 文本领域的数据预处理技术
  - 文本纠错
  - 低频词处理
  - 停用词
  - 关键词提取
  - 短语提取
  - 文本数据增强
- 深度学习训练技巧
  - 学习率调整
  - 梯度修正
  - 参数初始化
  - 归一化

NeuHub AI Links All

文本纠错技术

- 文本纠错又称为拼写错误或者拼写检查
- 一般存在两大纠错类型：
  - Non-word拼写错误
  - Real-word拼写错误

AI Links All

文本纠错技术：Non-word拼写错误

- 表示此词汇本身在字典中不存在，比如把“要求”误写为“药求”，把“correction”误拼写为“corrction”。

- 操作步骤
  - 找到候选词
  - 接着基于特定算法找出与错拼词关联最高的一个或多个单词作为纠正选项

- 如何确定候选项呢？
  - 英文：编辑距离
  - 中文：拼写相近、字形相近

NeuHub AI Links All

文本纠错技术：Non-word拼写错误

- 如何找出最关联项：

可以根据贝叶斯定理，得到如下表达式：

P（候选项|错误单词）∝ P（候选项）* P（错误单词|候选项）

- “此选项本身在语料中出现的可能性”和“人们意图打候选项时会错打成错误单词的可能性”的乘积。

- 前者可视为uni-gram语言模型，需要计算词语的出现频次，当然也可以扩张至二阶或三阶，比如计算“错误单词左边单词+候选项+错误单词右边单词”在语料中的出现情况。

文本纠错技术：Real-word拼写错误

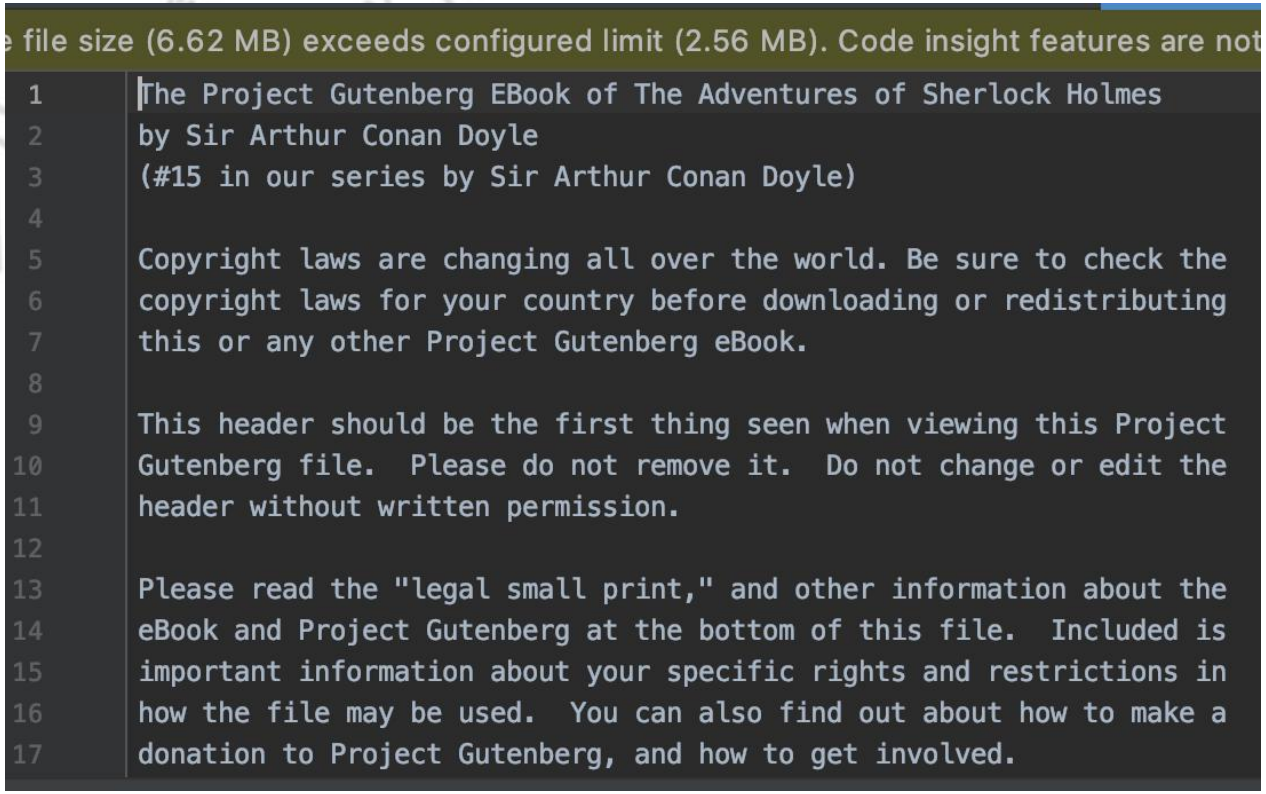
- 单词本身没有错误，但是不符合上下文语境，常常涉及语法语义层面的错误，比如把“我现在在公司里”错写成“我现在在公式里”。

- 操作步骤
  - 首先针对每个单词根据编辑距离、同音词、近形词等方式选出候选项（也包括单词本身）
  - 接下来计算基于候选项的语言模型，以及在候选项情况下出现错误单词的条件概率；如果综合计算而得单词本身出现在此语境中的概率较大，则不进行纠正，否则推荐纠正项。

- 注：在实际应用中不可能对每个词进行排查，可以应用语言模型等方式对句子进行粗粒度的初步筛查。

NeuHub Neu Links AII

作业：文本纠错技术：Non-word英文纠错



只给定少量英语语料，如何设计一个简单的英文纠错系统？

NeuHub
AI Links All

作业：文本纠错技术： Non-word英文纠错

只给定少量英语语料，如何设计一个简单的Non-word英文纠错系统？

候选项：
基于编辑距离 ， 如何找？

筛选：

- 语言模型（一阶 or 二阶）-> 统计
- 条件概率模型（无）

NeuHub AI Links All

词汇预处理：低频词

- 往往在NLP的应用中，我们通常先把出现频率很低的词汇过滤掉
- 原因
  - 出现频率特别低的词汇对语料整体的影响不大
  - 极大地减少计算复杂度（减小词典单词量）

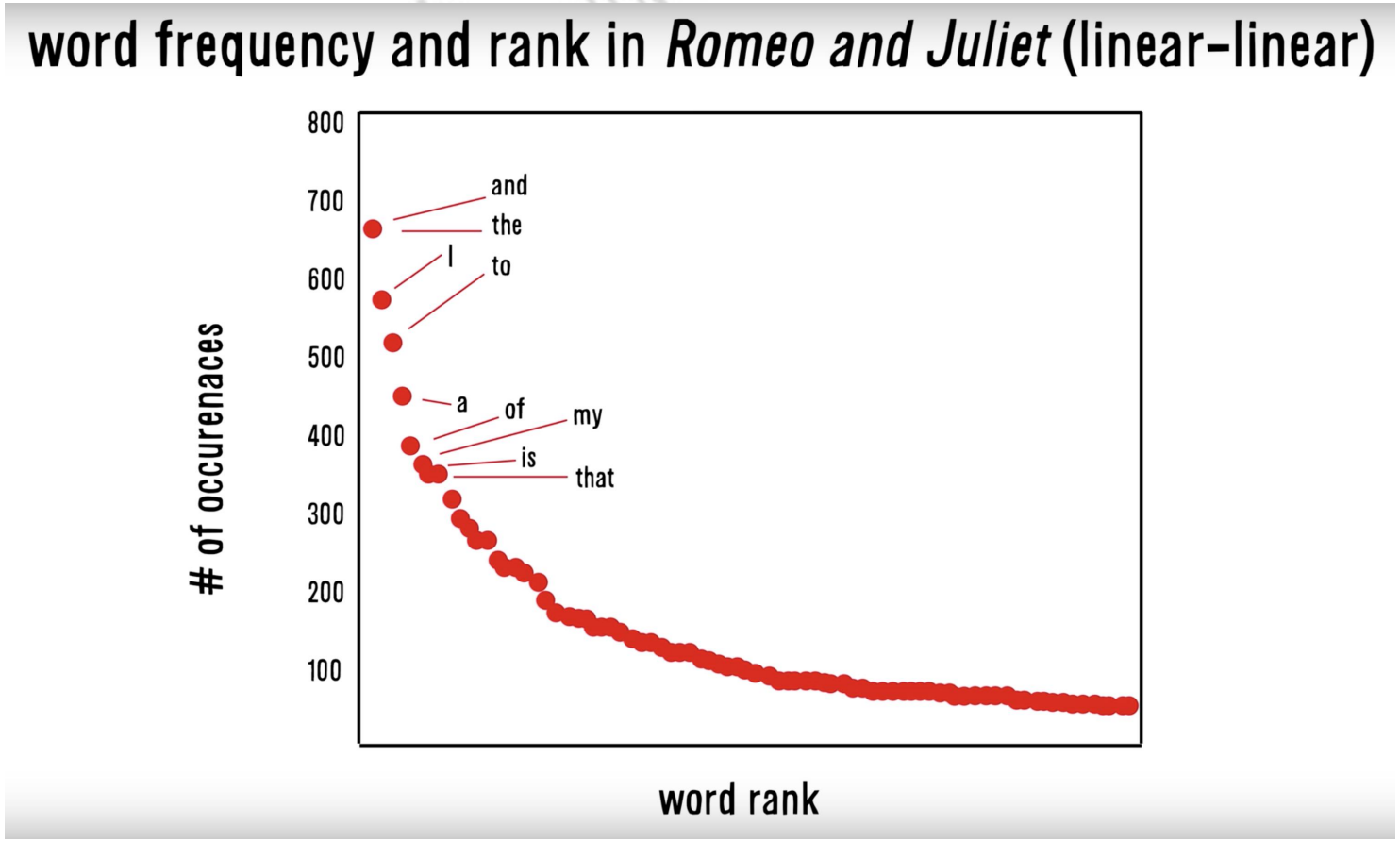
NeuHub AI Links All

词汇预处理：齐夫定律

为什么?

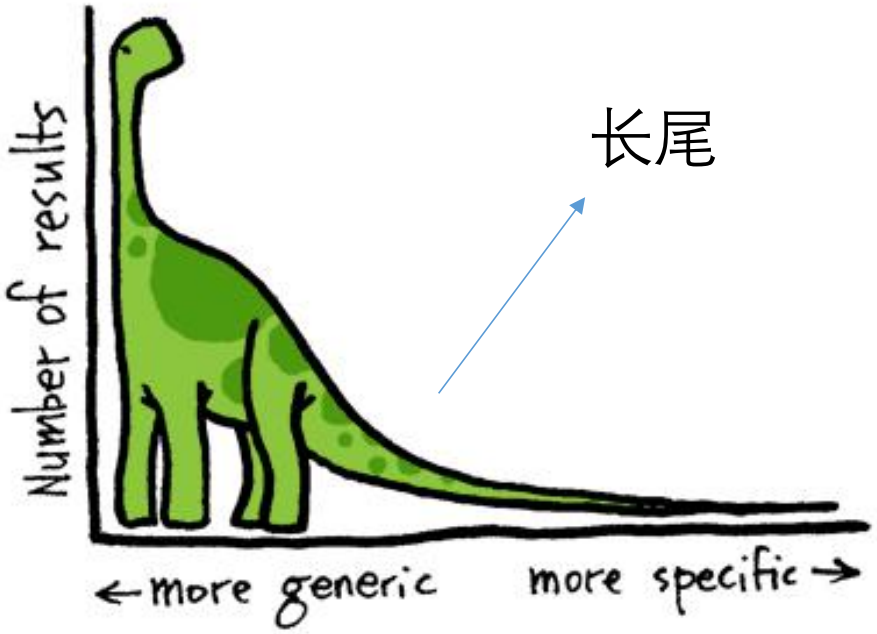
- 齐夫定律（英语：Zipf's law，IPA/ˈzɪf/）是由哈佛大学的语言学家乔治·金斯利·齐夫（George Kingsley Zipf）于1949年发表的实验定律。
- 在自然语言的语料库里，一个单词出现的频率与它在频率表里的排名成反比。

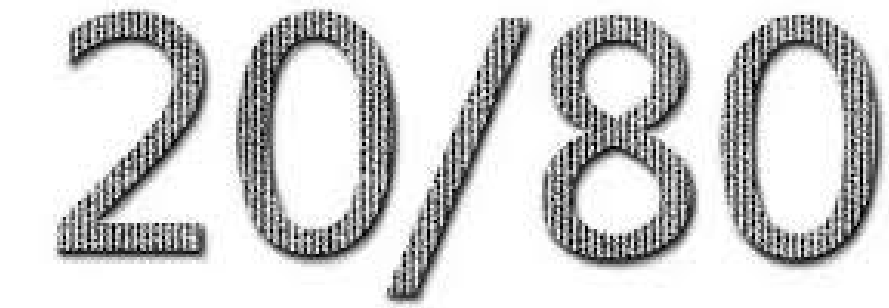
词汇预处理：齐夫定律


word frequency and rank in *Romeo and Juliet* (linear-linear)
# of occurenaces
word rank
800
700
600
500
400
300
200
100
and
the
I
to
a
of
my
is
that

CNR
A SILVA SCREEN Label
Soundtrack Music from
Franco Zeffirelli's
STEREO
DIGITALLY
MASTERED
'Romeo & Juliet'
Music from the
Academy Award
Winning Film
Composed
and Conducted by
Nino Rota
CONTAINS BOOKLET WITH COLOUR STILLS

词汇预处理：长尾现象，二八定律





NeuHC AI Lin All

词汇预处理：长尾现象，二八定律

中文词汇有几十万，常用的就几千个

- 少部分频率高的词汇占据了语料的大部分内容
- 大部分的单词出现频率低

- 出现频率特别低的词汇对语料整体的影响不大
- 极大地减少计算复杂度（减小词典单词量）

NeuHub
AI Links All

词汇预处理：停用词

京东智联云 & AI

语气词、助词等都可以作为停用词来处理。但是，也需要考虑自己的应用场景。

类似于特征筛选的过程

NeuHub AI Links All

词汇预处理：停用词

贪心科技 GREEDY TECHNOLOGY

情感分类

人物关系信息抽取

事件识别

“啊、吗、太、可”等语气助词也很重要

动词反而不那么重要，名词更重要

名词、动词都很重要

NeuHub AI Links All

关键词的提取：基于特征统计

- 词频：一般来说，一个词在文本中出现次数越多，表明作者越想表达这个词，因此可以通过对词频的简单统计便可以评估出词语的重要性。
- TF-IDF（term frequency – inverse document frequency）：综合考虑了词在文本中的词频以及普遍重要性，直观地说，一个词在其它文本中出现越少，在本文本中出现越多，那么对此文本的重要性越强。
- 位置特征
- 词跨度：指的是一个词在文本中首次与末次出现的距离
- 词的固有属性：包括一系列特征，词长、词性、对应的句法成分、开头大小写、是否全部大小写、词缀等。

NeuHub
AI Links All

关键词的提取：基于主题模型

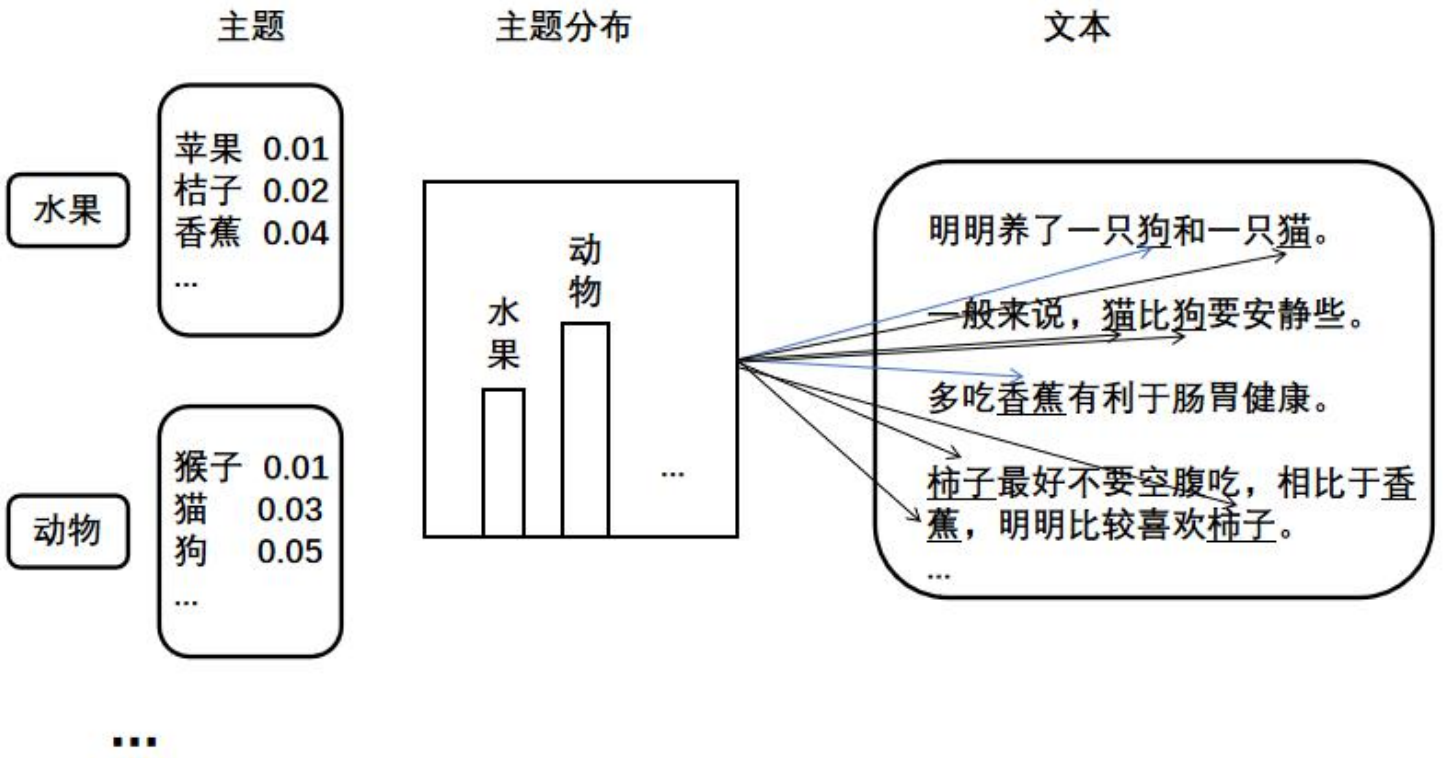
先观察这样几句文本：

- 明明养了一只狗和一只猫。
- 一般来说，猫比狗要安静些。
- 多吃香蕉有利于肠胃健康。
- 柿子最好不要空腹吃，相比于香蕉，明明比较喜欢柿子。

主题模型的核心假设是，存在隐含变量，即文本主题，决定了文本中词汇的出现情况。

NeuHub
AI Links All

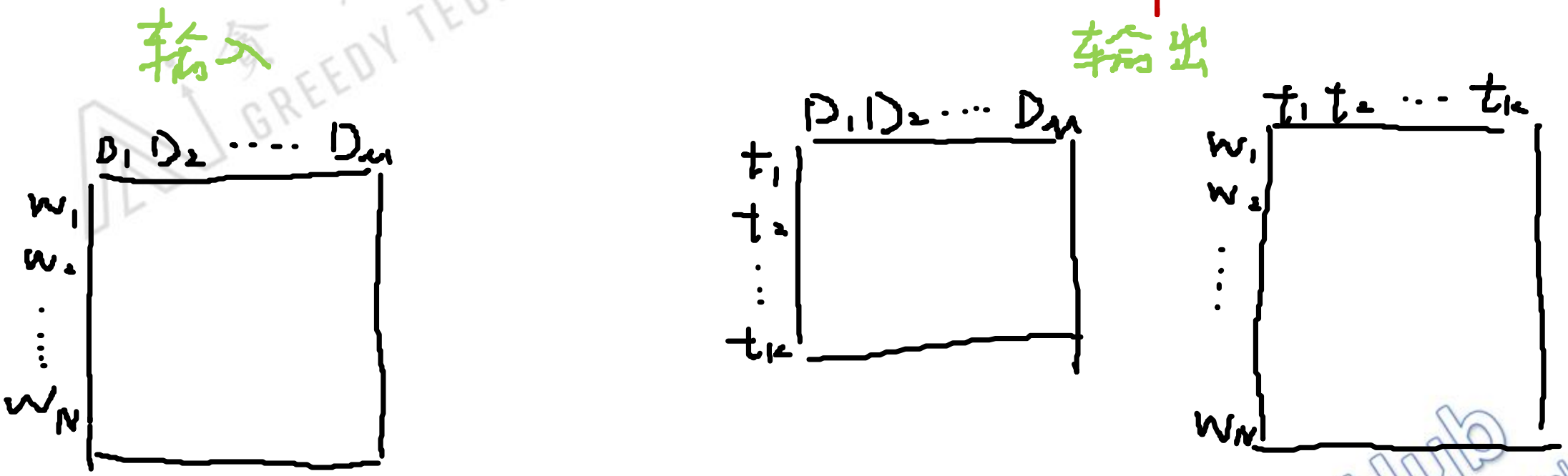
关键词的提取：基于主题模型



- LDA（Latent Dirichlet Allocation）是最典型的主题模型。
- 它假设语料库中蕴含了多个主题，而每个主题下面对应了一系列的词语。当作者在书写某篇文章时，是先设定某个主题再从中选择某个词的过程。

NeuHub
AI Links All

关键词的提取：基于主题模型



- 在具体运算中，首先需要指定主题的数目，一般需要根据实验结果多次调节，主题模型根据文本中词的出现频率情况即词袋模型去学习两类概率分布。
- 每个主题下概率较大的词汇可认为是关键词。

NeuHub All Links All

关键词的提取：基于主题模型

- 假设有M篇文章，K个主题，N个单词
- 目标：
  - 文档-主题分布：M个维度为K的多项式分布
  - 主题-单词分布：K个维度为N的多项式分布

- 多项式分布的参数服从Dirichlet分布（先验分布）。
- $f(\underset{p}{\rightarrow}) = \begin{cases} \frac{1}{B(\underset{\alpha}{\rightarrow})} \prod_{k=1}^{K} p_k^{\alpha_k - 1} \\ 0, \text{其它情况} \end{cases}$，$p_k \epsilon [0,1]$，$\sum P_k = 1$，$\alpha_k > 0$

- 而Dirichlet分布是多项式分布的共轭先验分布。
- 后验分布（Dirichlet分布） = 似然分布（多项式分布）*先验分布（Dirichlet分布）

- 初始化先验分布，迭代地求出最终的似然分布（多项式分布）

NeuHub AI Links All

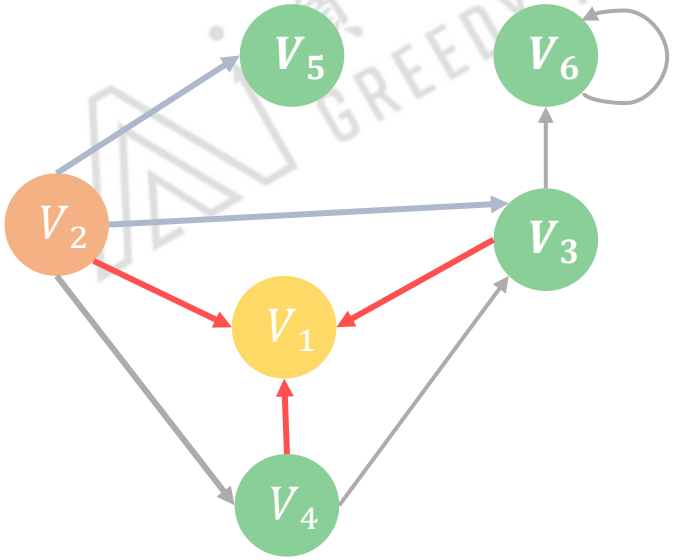
关键词的提取：由PageRank说起

PageRank

- PageRank，网页排名，Google用它来体现网页的相关性和重要性，在搜索引擎优化操作中是经常被用来评估网页优化的成效因素之一。
- 要点：
  - 如果一个网页被很多其他网页链接到，说明这个网页比较重要，也就是PageRank值会相对较高。
  - 如果一个PageRank值很高的网页链接到一个其他的网页，那么被链接到的网页的PageRank值会相应地因此而提高。

NeuHub AI Links All

关键词的提取：由PageRank说起

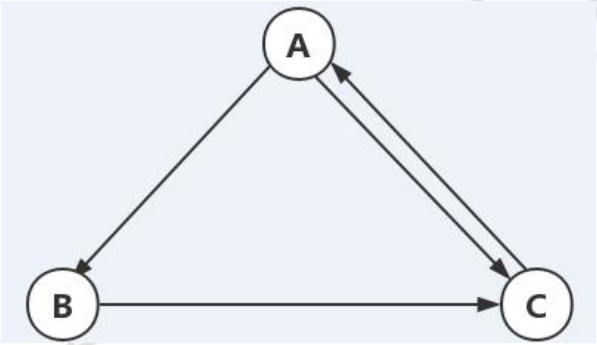


- 节点：网页
- 有向边：网页 *i* 到网页 *j* 的超链接

$$PR(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

- V_i ：某个网页
- V_j ：链接到V_i 的网页，即V_i 的入链
- $PR(V_i)$：网页V_i 的值
- $In(V_i)$：链接到V_i的节点集合
- $|Out(V_j)|$：从V_j出发链接到的节点个数，即V_j 的出链
- d：阻尼系数，其意义是，在任意时刻，用户到达某页面后并继续向后浏览的概率。通常d=0.85

关键词的提取：由PageRank说起



为了便于计算，我们假设每个页面的初始值为1，d为0.5。

$$PR(A) = 0.5 + 0.5 \times PR(C) = 0.5 + 0.5 \times 1 = 1$$

$$PR(B) = 0.5 + 0.5 \times (PR(A)/2) = 0.5 + 0.5 \times (1/2) = 0.75$$

$$PR(C) = 0.5 + 0.5((PR(A)/2) + PR(B)) = 0.5 + 0.5 \times (1/2 + 0.75) = 1.125$$

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

什么时候，迭代结束？

一般要设置收敛条件：上次迭代结果与本次迭代结果小于某个误差；最大循环次数。

$$S(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

关键词的提取：TextRank

京东智联云 & AI GREEDY TECHNOLOGY

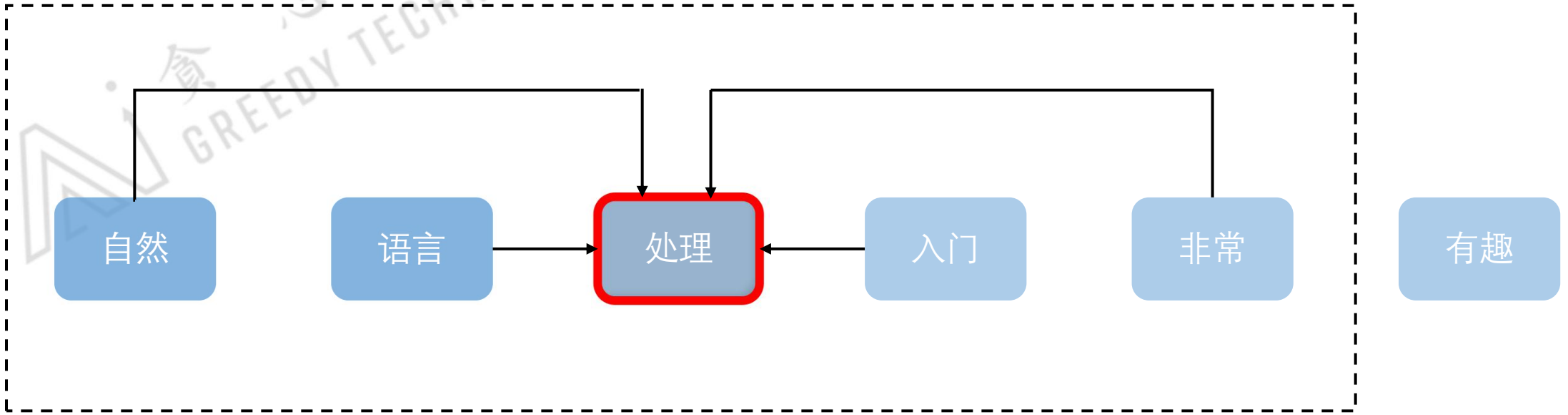
TextRank是PageRank在文本上的应用

核心思想如下：

- 如果一个单词出现在很多单词边上，说明这个单词比较重要。
- 一个TextRank值很高的单词边上的单词，TextRank值会相应地因此而提高。

NeuHub AI Links All

关键词的提取：基于TextRank

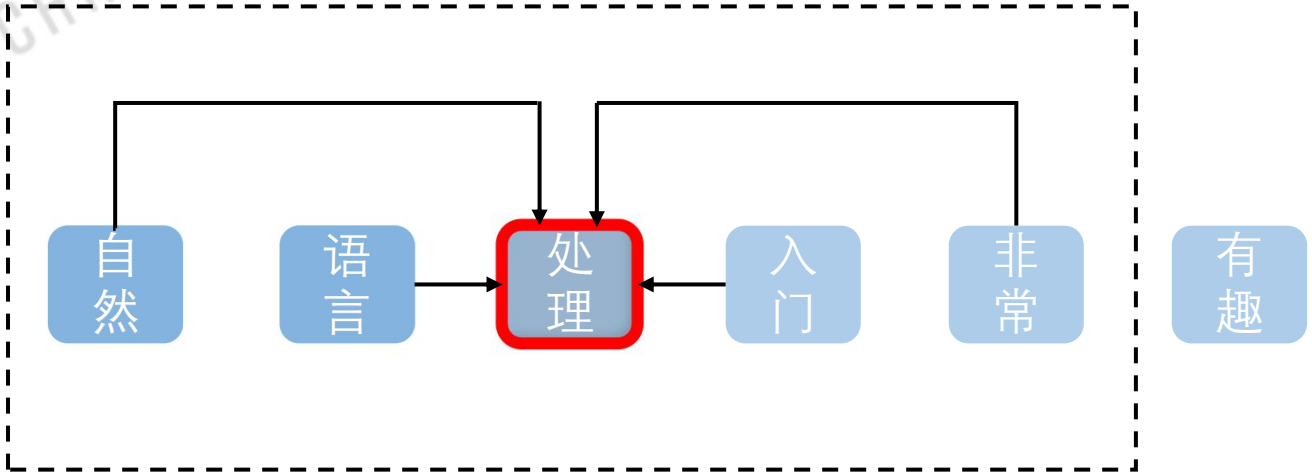


窗口半径为2

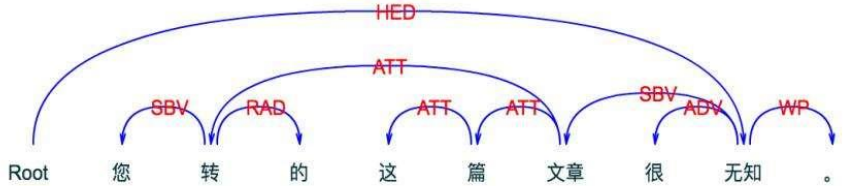
NeuHub
AI Links All

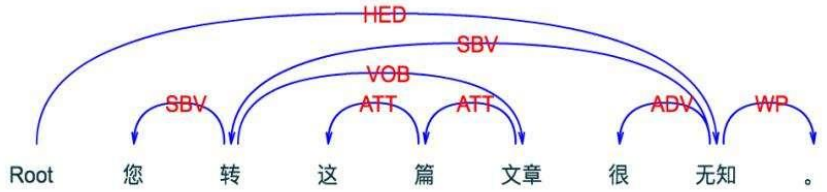
关键词的提取：基于TextRank

窗口半径为2



思考：改进点？
如：从句法分析的角度定义连接关系。





短语提取

- 固定的词汇搭配：“green hand”、“white lie”、“十八般武艺”、“鲤鱼跃龙门”
- 专业表达：“支持向量机”、“弱监督学习”、“自适应学习”
- 新的短语表达：“5G元年”、“基层减负年”、“雨女无瓜”

NeuHub AI Links All

短语提取步骤

1）基于N-gram统计获取出现频率较高的短语作为候选项。
2）对候选项进行多维度的特征统计。
3）将多维度特征进行综合评估、排序，取topK。

NeuHub
AI Links All

短语提取：特征指标

PMI（Pointwise Mutual Information）：中文译作点互信息，能够刻画单词间经常一起搭配的程度。

$$\mathbf{PMI}(w_i w_j) = \log\left(p(w_i w_j) \big/ p(w_i)p(w_j)\right)$$

两者联合出现的情况越多，单独出现的情况越少，此候选项为某一固定搭配的完整语义单元的可能性越大。
例：“支持向量机”，“支持”，“向量机”

NeuHub AI Links All

短语提取：特征指标

左（右）邻字熵：如果是某一完整意义的短语，其左右的搭配应当很丰富，此指标从熵的角度来衡量某一候选项的自由搭配程度。

$$P(w_i w_j) = -\sum_{w \in W} p(w) \log p(w)$$

其中W表示候选项左（右）侧所出现字的集合，此集合越是丰富，说明候选项为某固定短语的可能性越高。

例：“持向量机” 左侧：“支”

“支持向量机”左侧：字多种多样

NeuHub AI Links All

短语提取->新词挖掘

延伸：基于点互信息、左（右）邻字熵进行新词挖掘

以字为单位进行操作

https://github.com/lujiaying/BaizeNLP/tree/master/worddiscovery

《西游记》

行者，八戒，师父，三藏，行者道，大圣，一个，唐僧，
菩萨，沙僧，和尚，怎么，者道，我们，不知，长老，
那里，笑道，妖精，老孙，悟空，甚么，两个，八戒道，
国王，徒弟，闻言，那怪，如何，呆子，只见，三藏道，与他，
不敢，不曾，宝贝，小妖，原来，大王，道师父，今日，正是，
等我，兄弟，出来，叫道，如今，一声，取经，铁棒

文本数据增强

针对文本数据，在未转化成数值表征形式的时候，可以在字符层面进行数据增广，常见的方法有：

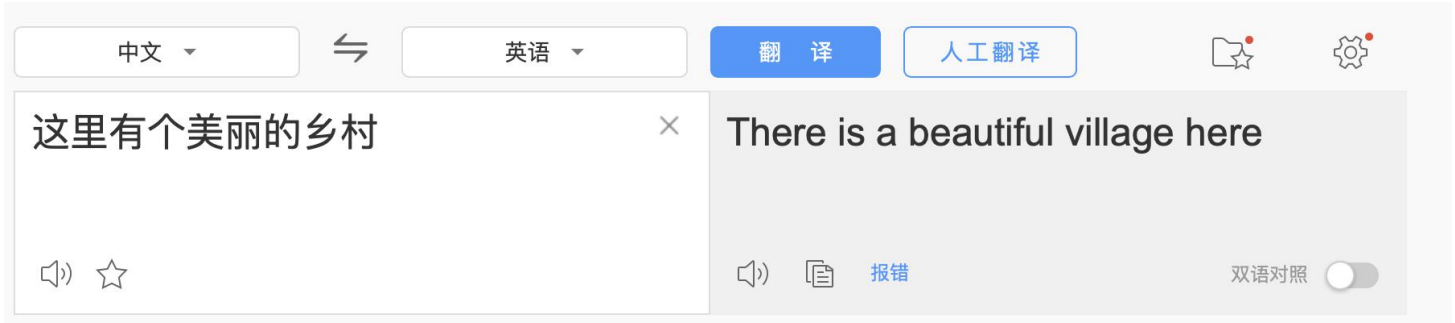
- 同义词替换

这里有个美丽的乡村。-> 这里有个美丽的村庄。

NeuHub AI Links All

文本数据增强

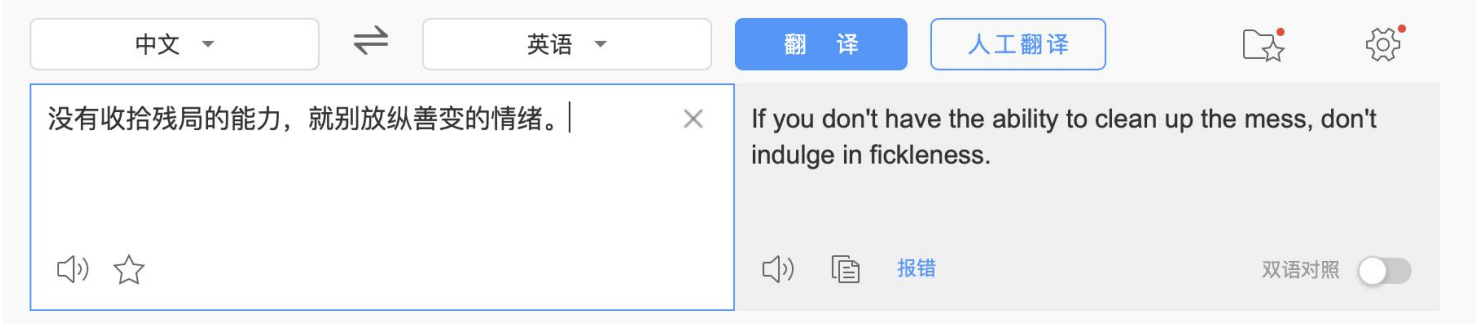
- 文本回译

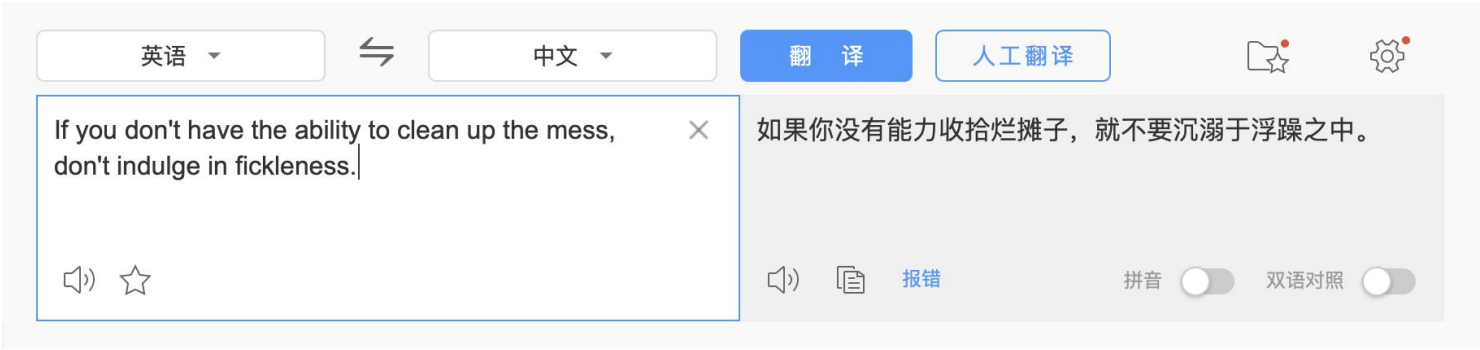


AI Links All

文本数据增强

- 文本回译





AI Links All

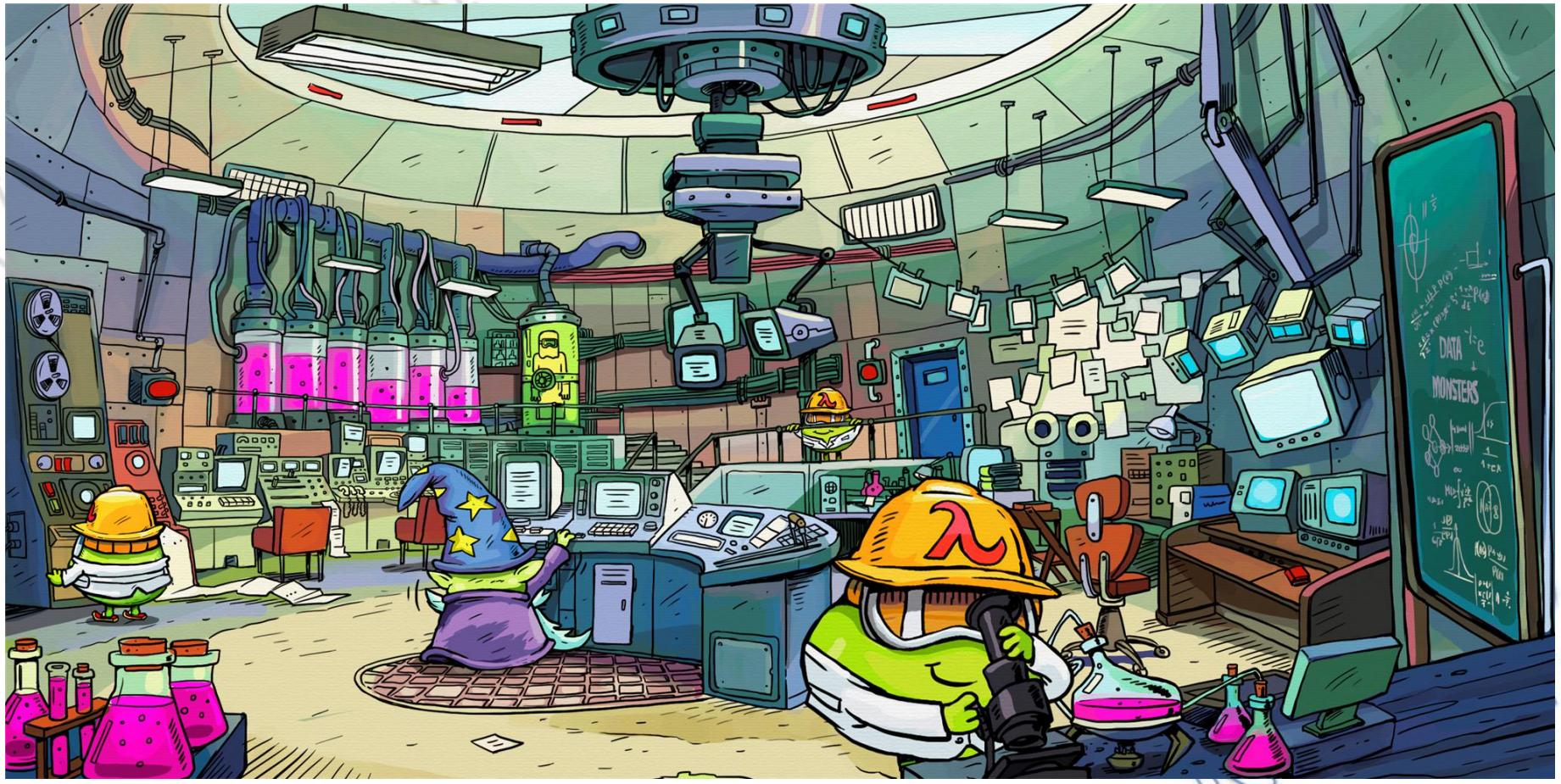
文本数据增强

- 转换顺序：比如文本分类任务，对语序不敏感

- 一键中文数据增强工具：https://github.com/425776024/nlpcda

NeuHub
AI Links All

休息



京东智联云 & 贪心科技 GREEDY TECHNOLOGY
Links AI

神经网络模型的问题

京东智联云 &

➢优化问题：

- 神经网络的损失函数为非凸函数，找到全局最优解通常比较困难。
- 神经网络有大量参数，训练数据也很大量，一阶优化方法的训练效率通常比较低，而二阶优化方法计算代价很高。
- 深度神经网络存在梯度消失或爆炸问题，导致基于梯度的优化方法经常失效。

➢泛化问题：

- 由于深度神经网络的复杂度比较高，并且拟合能力强，很容易在训练集上产生过拟合。因此在训练深度神经网络时，同时也需要通过一定的正则化方法来改进网络的泛化能力。

NeuHub A LinkAI

神经网络模型的问题

- 梯度下降法方式
- 学习率调整
- 梯度修正
- 参数初始化
- 归一化

NeuHub
AI Links All

梯度下降法

- 深度神经网络的参数学习主要通过梯度下降法来寻找一组可以最小化结构风险的参数。
- 在具体实现中，梯度下降法可以分为：批量梯度下降（速度慢，稳定）、随机梯度下降（速度快，不稳定）以及小批量梯度下降。
- 基于速度与稳定性的考量，经常使用小批量梯度下降法。
- 批量大小如何选择？

NeuHub
AI Links All

梯度下降法

➢批量大小如何选择？

- 一般而言，批量大小会影响随机梯度的方差（即稳定性）。
- 批量越大，随机梯度的方差越小，引入的噪声也越小，训练也越稳定，相应地，可以设置较大的学习率。批量较小时，需要设置较小的学习率，否则会影响模型收敛。
- 学习率通常要随着批量大小的增大而相应地增大。
- 线性缩放规则：当批量大小增加 m倍时，学习率也增加m倍。

NeuHub AI Links All

学习率调整

- 在梯度下降法中，学习率α的取值很关键，如果过大很可能不会收敛，如果过小则收敛速度太慢。
- 学习率调整方法：
  - 学习率衰减
  - 学习率预热
  - 自适应调整学习： AdaGrad、RMSprop、AdaDelta

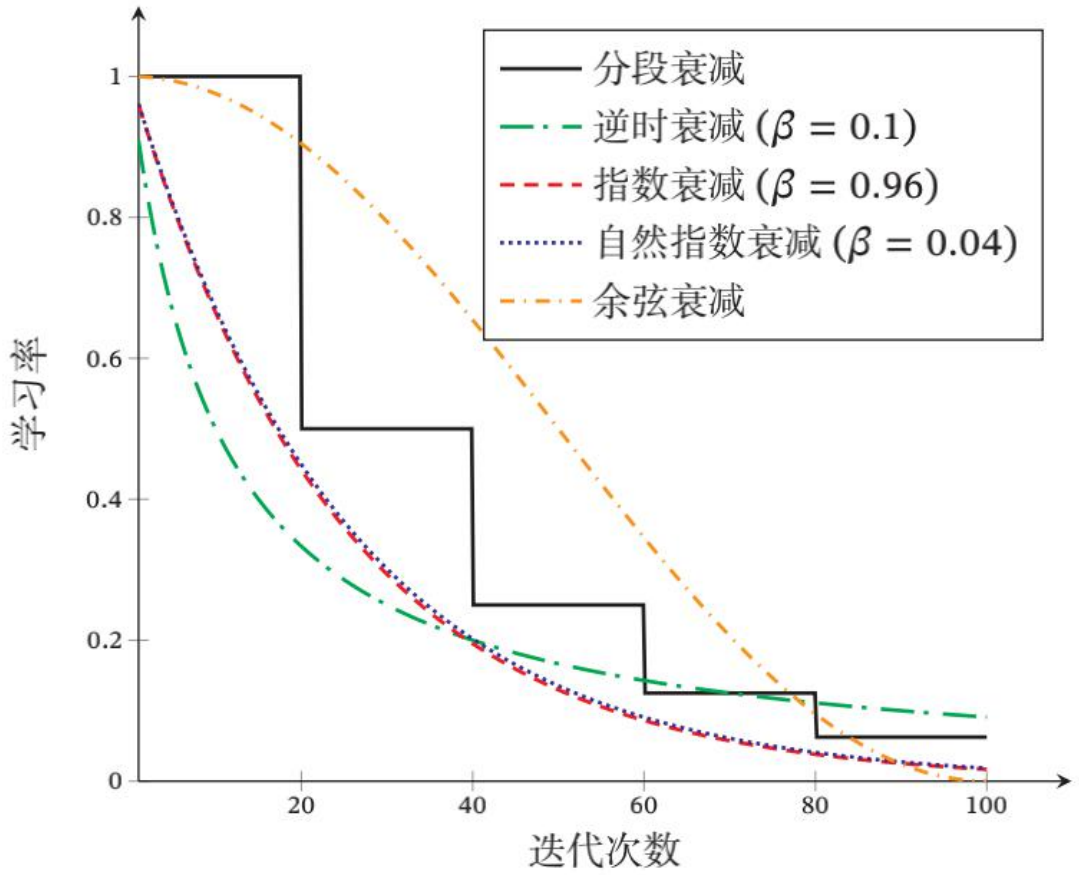
NeuHub AI Links All

学习率调整

➢学习率衰减

一开始可设置大点的学习率；

在最优点附近后减小学习率



$$\alpha t = \alpha 0(\frac{1}{1+\beta * t})$$

$$\alpha t = \alpha 0 * \beta^{t}$$

$$\alpha t = \alpha 0 * exp(-\beta * t)$$

$$\alpha t = 0.5 * \alpha 0(1 + cos(\frac{t\pi}{T}))$$

T表示总迭代次数

学习率调整

京东智联云 & AI

➢学习率预热

为了提高训练稳定性，在最初几轮迭代时，采用比较小的学习率，等梯度下降到一定程度后再恢复到初始的学习率，这种方法称为学习率预热。

NeuHub AI Links All

学习率调整

京东智联云 & AI

➢AdaGrad算法（Adaptive Gradient Algorithm）

- 在一般的梯度下降法中，每个参数在每次迭代时都使用相同的学习率。但实际上，在每个参数的维度上收敛速度都不相同，AdaGrad根据不同参数的收敛情况分别设置学习率。
- 在第t 次迭代时，先计算参数梯度平方的累计值：$Gt = \sum_{i=0}^{T} g_i^2$
- 参数更新差值$\Delta\theta_t$为（gt表示第t次迭代计算出的梯度，ε是一个常数，取exp（-7）到exp（-10）之间）：

$$-\frac{\alpha}{\sqrt{Gt+\varepsilon}}g_t$$

NeuHub AI Links All

学习率调整

京东智联云 & 贪心科技

➢AdaGrad算法（Adaptive Gradient Algorithm）

- 参数更新差值$\Delta\theta_t$为（gt表示第t次迭代计算出的梯度，ε是一个常数，取exp（-7）到exp（-10）之间）：

$$-\frac{\alpha}{\sqrt{Gt+\varepsilon}}g_t$$

- 学习率将随着梯度的倒数增长，也就是说有较大梯度累积的具有较小的学习率，而较小的梯度累积的具有较大的学习率，可以解决普通的sgd方法中学习率一直不变的问题。
- 由于分母中对历史梯度一直累加，学习率将逐渐下降至0，在经过一定次数的迭代依然没有找到最优点时，由于这时的学习率已经非常小，很难再继续找到最优点。

AI Links All

学习率调整

京东智联云 & 贪心科技

➢RMSProp算法

- RMSProp优化算法和AdaGrad算法唯一的不同，在于累积平方梯度Gt的求法不同，用一个衰减系数β(一般取0.9)来控制历史信息的获取。
- 历史梯度信息的累计乘上一个衰减系数 β，然后用(1– β)作为当前梯度的平方加权系数相加。
- $Gt = \beta G_{t-1} + (1-\beta) g_t^2$
- 在迭代过程中，每个参数的学习率并不是呈衰减趋势，既可以变小也可以变大，在有些情况下可避免 AdaGrad 算法中学习率不断单调下降以至于过早衰减的缺点。

NeuHub AI Links All

学习率调整

- AdaDelta算法
- 不仅对历史梯度信息做有选择的累加，对历史的参数更新差值$\Delta\theta$也做有选择的累加。
- $\Delta X_{t-1} = \gamma \Delta X_{t-2} + (1-\gamma)\Delta\theta_{t-1}^2$
- 对于Gt的运算和RMSProp一样
- 参数更新差值$\Delta\theta_t$为：

$$-\frac{\sqrt{\Delta X_{t-1}+\varepsilon}}{\sqrt{Gt+\varepsilon}}g_t$$

在一定程度上保持了学习率的平稳性

NeuHub AI Links All

梯度修正

- 在梯度下降法中，除了学习率调整，还可以调整梯度。
- 批量梯度下降法中，如果每次选取样本数量比较少，损失会呈现振荡的方式下降（与整体训练集上的最优梯度不一致）。
- 解决方式：通过使用最近一段时间内的平均梯度来代替当前时刻的随机梯度来作为参数更新的方向，从而提高优化速度。
  - Momentum动量
  - Nesterov加速梯度度（Nesterov Accelerated Gradient，NAG）
  - Adam（Adaptive Moment Estimation Algorithm）

NeuHub AI Links All

梯度修正

➢Momentum动量法

- 是模拟物理中的概念，某物体的动量指的是该物体在它运动方向上保持运动的趋势。动量法（Momentum Method）是会考虑之前的梯度情况。
- 参数更新差值$\Delta\theta_t$为（p通常设为0.9）：

$$p\Delta\theta_{t-1} - \alpha g_t = -\alpha \sum_{i=1}^{t} p^{t-i} g_i$$

- 更新取决于最近一段时间内梯度的加权平均值。当某个参数在最近一段时间内的梯度方向不一致时，参数更新幅度变小；反之，参数更新幅度变大，起到加速作用。

NeuHelp AI Links All

梯度修正

京东智联云 &

➢Nesterov加速梯度

- 是对Momentum动量法的改进
- 在动量法中，参数更新方向$\Delta\theta t$ 为：上一步的参数更新方向$\Delta\ \theta t$-1和当前梯度的反方向$-\boldsymbol{g}t$ 的叠加。
- 相当于分为两步进行，先根据$\Delta\theta t$-1 更新一次得到参数θ'，再用-gt进行更新。

$$\theta' = \theta_{t-1} + \rho\Delta\theta_{t-1}$$

$$\theta_t = \theta' - \alpha\boldsymbol{g}t$$

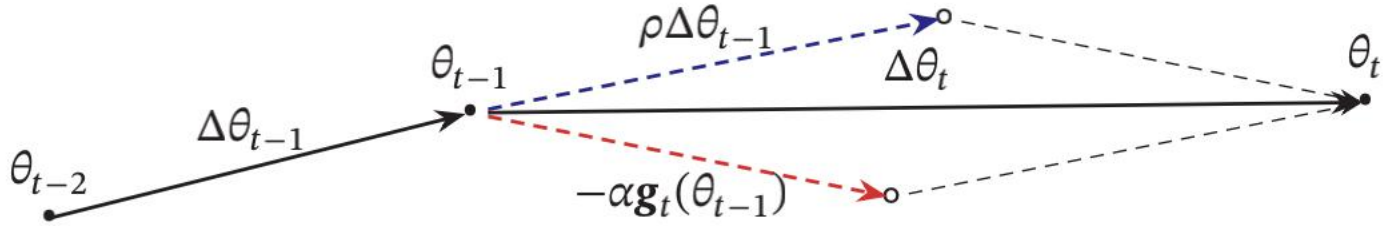
- 其中梯度$\boldsymbol{g}t$ 为点$\theta t-1$ 上的梯度，因此在第二步更新中有些不太合理，更合理的更新方向应该为θ' 上的梯度。

g't

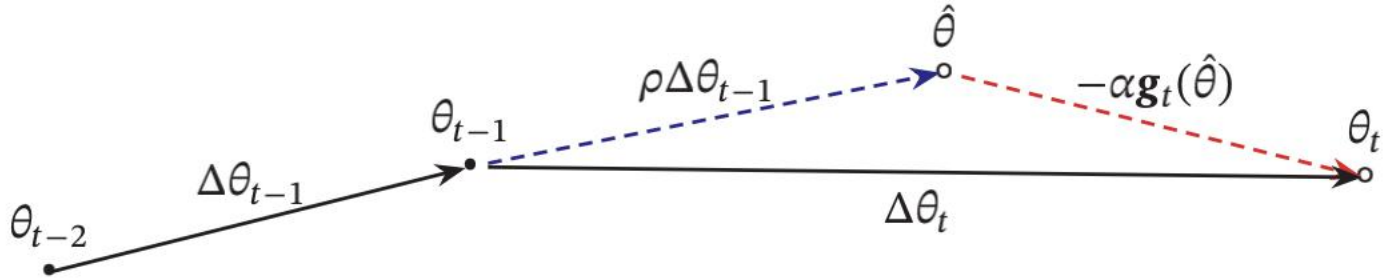
NeuHub AI Links All

梯度修正

- Nesterov加速梯度



(a) 动量法



(b) Nesterov 加速梯度

梯度修正

➢Adam算法

- Momentum和 RMSprop 算法的结合，不但使用动量作为参数更新方向，而且可以自适应调整学习率。

$$\text{RMSprop}：\ Gt = \beta G_{t-1} + (1-\beta) g_t^2$$

$$\text{Momentum}：\ Mt = \gamma M_{t-1} + (1-\gamma) g_t$$

- 通常β=0.9，λ=0.99

NeuHub AI Links All

梯度修正

京东智联云 & 贪心科技

➢Adam算法

- 假设$M0 = 0, G0 = 0$，那么在迭代初期Mt 和Gt 的值会比真实的均值和方差要小，进行修正：
- $G't = \frac{Gt}{1-\beta}$ $M't = \frac{Mt}{1-\gamma}$
- 参数更新差值$\Delta\theta_t$为：

$$-\frac{\alpha}{\sqrt{G't+\varepsilon}}M't$$

NeuHub AI Links All

总结

学习率调整	学习率衰减、学习率预热、自适应（AdaGrad, RMSprop, AdaDelta）
梯度修正	Momentum, Nesterov, Adam

NeuHub AI Links All

总结

original	$-\alpha g_t$
AdaGrad	对学习率进行调节，考虑历史梯度平方累积
RMSprop	对学习率进行调节，考虑历史梯度平方加权累积
AdaDelta	对学习率进行调节，考虑历史梯度平方加权累积以及历史更新梯度加权累积
Momentum	对梯度进行调节，考虑历史梯度累积
Nesterov	对梯度进行调节，改进Momentum中的历史梯度累积
Adam	RMSprop+Momentum 历史梯度、历史梯度平方、历史更新

参数初始化

- 参数初始值的选取十分关键，关系到网络的优化效率和泛化能力。
- 一般而言，参数初始化的区间应该根据神经元的性质进行差异化的设置。
- 如果一个神经元的输入连接很多，它的每个输入连接上的权重就应该小一些，以避免神经元的输出过大（当激活函数为 ReLU 时）或过饱和（当激活函数为Sigmoid函数时）。

NeuHub
AI Links All

参数初始化

- 预训练初始化：通常情况下，一个已经在大规模数据上训练过的模型可以提供一个好的参数初始值，这种初始化方法称为预训练初始化。
- 随机初始化：使得不同神经元之间的区分性更好。
- 如何选取合适的随机初始化区间？
- 为了使得训练平稳性，应当尽量保持每个神经元的输入和输出的方差一致。

NeuHub
AI Links All

参数初始化

京东智联云 & AI

➢ Xavier初始化

- 在神经网络中，第 l 层的一个神经元值 $a^{(l)}$，接收到前一层的 M_{l-1}个神经元的输出$a_i^{(l-1)}$(这里令激活函数$f(\cdot)$为恒等函数)：

$$a^{(l)} = f(\sum_{i=1}^{M_{l-1}} w_i^{(l)} a_i^{(l-1)})$$

- $a^{(l)}$的方差为（假设w，a的均值为0，且独立地服从同样的分布）：

$$var(a^{(l)}) = var(\sum_{i=1}^{M_{l-1}} w_i^{(l)} a_i^{(l-1)}) = \sum_{i=1}^{M_{l-1}} var(w_i^{(l)}) var(a_i^{(l-1)})$$

$$= M_{l-1} var(w_i^{(l)}) var(a_i^{(l-1)})$$

NeuHub AI Links All

参数初始化

京东智联云 & 贪心科技

- Xavier初始化

- $a^{(l)}$的方差为：

$$var(a^{(l)}) = M_{l-1} var(w_i^{(l)}) var(a_i^{(l-1)})$$

$$令 var(w_i^{(l)}) = 1/M_{l-1}$$

考虑到在反向传播中，误差信号也不被放大或缩小：

$$令 var(w_i^{(l)}) = 1/M_l$$

综合两者：

$$令 var(w_i^{(l)}) = 2/（M_{l-1} + M_l）$$

NeuHub AI Links All

参数初始化

- Xavier初始化

综合两者：

$$令var(w_i^{(l)}) = 2/（M_{l-1} + M_l）$$

注：这里假设激活函数为恒等函数，但是 也适用于 Logistic 函数和 Tanh 函数。因为神经元的参数和输入的绝对值通常比较小，处于激活函数的线性区间，这时Logistic函数和Tanh函数可以近似为线性函数。在实际应用中，通常乘以一个缩放因子ρ。

NeuHub AI Links All

归一化

- 将传统机器学习中的数据归一化方法应用到深度神经网络中，对神经网络中隐藏层的输入进行归一化，从而使得网络更容易训练。
- Why：某一神经层的输入分布会受到前几层参数变化的影响（越是后面的层，分布偏移越大）；而每次参数更新都会导致神经层的输入分布在不同批次间也发生改变。从机器学习角度来看，如果神经层的输入分布发生了改变，那么其参数需要重新学习（Internal Covariate Shift）。
- How：对神经层中间过程进行归一化操作，使分布保持稳定。

NeuHub AI Links All

归一化

> *Training Deep Neural Networks is complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This slows down the training by requiring lower learning rates and careful parameter initialization, and makes it notoriously hard to train models with saturating nonlinearities.*

— Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015.

归一化

- 批归一化
- 对于神经网络中的每一个神经元（线性变化之后，非线性变化之前），基于Mini-batch进行标准归一化。
- 但是：取值集中到 0 附近，若使用 Sigmoid型激活函数时，此区间接近线性变换区间，减弱了神经网络的拟合能力。
- 因此：为了避免对神经网络的表示能力造成负面影响，增加两个参数用于分布的缩放和平移。

NeuHub AI Links All

归一化

- 批归一化

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

归一化

京东智联云 & AI

➢批归一化特点

- 批归一化可以看作一个特殊的神经层：非线性变化->批归一化->线性变化
- 小批量数据的时候，噪声较大。
- 当训练完成时，用整个数据集上的均值与方差用来作归一化。
- 相当于正则化方法：神经网络在训练时，对一个样本的预测不仅考虑到该样本本身，也考虑到同一批次中的其他样本，因此使得神经网络不会“过拟合”到某个特定样本，从而提高网络的泛化能力。

NeuHub AI Links All

归一化

京东智联云 & 贪心科技 GREEDY TECHNOLOGY

➢为什么批归一化有效果？

争议性：

- the original explanation of "internal covariate shift" is not very precise, so at best it is a heuristic explanation, and at worst it might be completely wrong：
  https://arxiv.org/pdf/1807.03341.pdf
- success of batch normalization has nothing to do with internal covariate shift：
  https://arxiv.org/pdf/1807.03341.pdf

NeuHub AI Links All

归一化

京东智联云 & 贪心科技

➢批归一化优点：

- 可应用大学习率
- 对权重初始化不太敏感
- 替代数据preparation
- 可省略Dropout

NeuHub AI Links All

归一化

京东智联云 & AI

➢批归一化缺陷：

- 对单个神经元进行归一化操作，因此要求小批量样本的数量不能太小，否则与整体数据的均值与方差偏移太大。

NeuHub AI Links All

归一化

➢层归一化：

- 批量归一化是不同训练数据之间对单个神经元的归一化，层归一化是单个训练数据对某一层所有神经元之间的归一化。（即均值与方差的计算方式不同）。

NeuHub
AI Links All

归一化

贪心科技 GREEDY TECHNOLOGY
京东智联云 & 贪心科技
NeuHub AI Links All

参考链接

深度学习，邱教授
https://nndl.github.io/

归一化
https://mlexplained.com/2018/11/30/an-overview-of-normalization-methods-in-deep-learning/
https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/

NeuHub
AI Links All